old line numbering

501.93144X00
filed 9/17/03

DATA TRANSFER METHOD

BACKGROUND OF THE INVENTION

~~Field of the Invention~~

The present invention relates to a virtualization technique for virtualizing the use of memory areas of plural memory devices; more particularly, the invention relates to a method of transferring data among memory devices or in a memory device.

~~Description of the Related Art~~

In a Storage Area Network (SAN), which employs a technique for connecting plural computers and plural memory devices through a network, there is a technique for virtualizing the use of memory areas of the memory devices (e.g., see "Virtualization of Disk Storage (September 2000)", which is a technology white paper of the Evaluator Group, Inc. ~~This literature is referred to as a cited literature in this specification~~).

There are several methods of virtualization depending upon the position between a computer and a memory device where virtualization is performed.

A first method is a method ~~with~~ which virtualization is performed by storage management software or volume management software in a computer in which a server application is executed.

A second method is a method ~~with~~ which a computer having plural interfaces connecting a memory device is placed in front of the memory device to perform virtualization, or the memory

device itself performs virtualization.

A third method is a method with which virtualization is performed in a network apparatus or a management server apparatus constituting the SAN. This method is classified into an in-band method of handling a control I/O (Input/Output) and a data I/O in the same network and an out-of-band method of separately providing a network for flowing the control I/O, to handle the control I/O separately from the data I/O.

SUMMARY OF THE INVENTION

In the second or third method of virtualization described above in the related art, a virtualized memory area (virtual volume) is provided by a relay device, such as a LAN switch, in the case of constituting an SAN using an Internet small computer system interface (iSCSI) in a fiber channel switch or the Ethernet (registered trademark), and an apparatus of a server base including plural host bus adaptors (HBAs) of a fiber channel or the Ethernet (registered trademark). In such a case, if the virtual volume is constituted by a combination of memory areas of plural memory devices, depending upon the state of access of the virtual volume, a memory device of a destination to be accessed changes one after another among the plural memory devices, for example, a first memory device, a second memory device, and a third memory device. As a result, there is a problem in that the relay processing load increases.

It is a first object of the present invention to provide a method of reducing the relay processing load by performing data transfer during operation, such that the virtual volume is constituted by memory areas of ~~the~~ a relatively small number of memory devices.

In addition, it is possible that, if the virtual volume is constituted by memory areas of plural memory devices in order to increase or decrease areas whenever necessary, the number of unused memory areas with a relatively small capacity may increase. If the virtual volume is constituted by a large number of memory areas with a small capacity, the processing load of conversion processing in associating the virtual volume in the relay device with an actual memory area increases.

Further, there is also a problem in that a relatively large virtual volume cannot be constituted by ~~the~~ a relatively small number of unused memory areas.

A second object of the present invention is to provide a method of reducing this problem by performing data transfer during operation, for example, from one or more memory areas already constituting the virtual volume to another memory area, such that the number of unused memory areas with a relatively small capacity ~~are~~ is reduced.

The present invention employs a data transfer method in a computer system including: plural computers; plural memory devices; a relay device which connects the computers and the

memory devices; and a management device which manages the computers, the memory devices, and the relay device,

wherein the management device sets virtual memory areas of the memory devices for the plural computers and holds information on contents of the setting as first information,

the relay device holds second information which is created based upon the first information,

the virtual memory areas correspond to memory areas in ~~the~~ respective memory areas or a memory area formed by combining memory areas in the plural memory devices, and

the relay device selects one virtual memory area from the second information and, ~~with~~ in the case ~~in which~~ where the selected virtual memory area is a memory area formed by combining the memory areas in the plural memory devices as an opportunity for data transfer, performs data transfer among the plural memory devices to reduce the number of memory devices involved in the combination.

In addition, the relay device refers to ~~the~~ second information and, ~~with~~ in the case in which the number of memory areas with a relatively small capacity not corresponding to the virtual memory areas increases as an opportunity for data transfer, performs data transfer among the plural memory devices to reduce the number of memory areas with a relatively small capacity not corresponding to the virtual memory areas.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram showing an example of the structure of a computer system;

Fig. 2 is a block diagram showing an example of the structure of programs and tables which a switch manager of a switch has;

Fig. 3 is a block diagram showing an example of the structure of programs and tables which a routing control section of the switch has;

Fig. 4 is a block diagram showing an example of the structure of programs and tables which a cooperative server of the switch has;

Fig. 5 is a block diagram showing an example of the structure of programs and tables which a management device of the computer system has;

Fig. 6 is a block diagram showing an example of the structure of a unit and an example of the structure of a unit management table;

Fig. 7 is a block diagram showing an example of the structure of a logical unit management table and a virtual volume management table;

Fig. 8 is a block diagram showing an example of the structure of the logical unit management table and the virtual volume management table;

Fig. 9 is a block diagram showing an outline of the process flow of data transfer;

Fig. 10 is a process flow diagram showing an example of a processing

sequence for obtaining an opportunity for data transfer;

Fig. 11 is a process flow diagram showing an example of the processing sequence of data transfer;

Fig. 12 is a diagram showing an example of a table for managing a state of a logical unit;

Fig. 13 is a diagram showing an example of a block bitmap table and an I/O processing matrix for performing I/O processing during data transfer;

Fig. 14 is a block diagram showing an outline of the process of data transfer;

Fig. 15 is a process flow diagram showing an example of a processing sequence in the case in which easiness of constitution of a disk is taken as an opportunity for data transfer;

Fig. 16 is a diagram showing an example of the structure of a unit usage state management table;

Fig. 17 is a block diagram showing an example of the structure in the case in which main components of a switch are constituted redundantly;

Fig. 18 is a block diagram showing a method of distribution and synchronization of table information of switch managers constituted redundantly; and

Fig. 19 is a block diagram showing an example of a structure in which the switch manager has a hard disk.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

A first embodiment of the present invention will be described with reference to Figs. 1 to 13.

In the first embodiment, a case will be described in which data transfer is performed during operation such that the virtual volume is constituted by memory areas of the relatively small number of memory devices.

Fig. 1 is a diagram showing an example of a structure of a computer system in the present invention. Server apparatuses 170 are connected to storage devices 180 via a switch 100.

Although the switch 100 is a fiber channel switch or a LAN switch, an Ethernet (registered trademark) switch is described as an example of a relay device in the description of embodiments of the present invention.

The switch 100 includes: plural connection interfaces 110 (represented as Ether MACs in Fig. 1) which perform control of a physical layer and a data link layer for connecting the server apparatuses 170, storage devices 180, or the like; plural routing control sections (routing controls) 120 which perform determination of a connection interface to be a destination and conversion of contents of a packet, if necessary, based upon information of a header or an information part of a layer 2 or upper layers of a packet to be sent and received (layers up to a layer 7 are shown in the figure); a crossbar switch 130 which connects the plural routing controls; a switch

management section (switch manager) 140 which performs device management of the switch 100 and calculation of a path control protocol; and an internal communication line 101C, such as an internal bus, which connects the crossbar switch 130 and the switch manager 140.

The routing control 120 includes: a CPU 126; a main memory (MEM) 127 for storing programs and tables; a forwarding control section ~~of~~ for forwarding packets from the layer 2 to the layer 7 (layer 2-7 forwarding engine) 121; a header search section (search engine) 122 for a packet; and a memory (MEM) 123 for storing a table 125 ~~thereof~~. These components are connected by an internal communication line 128, such as an internal bus.

The switch manager 140 includes: a CPU 141; a main memory (MEM) 142 for storing programs and a table 143, such as a path control table; a management Ethernet (registered trademark) (Ether) 145; and an internal bus control 144 for making connection via ~~with~~ the internal communication line 101C to the crossbar switch 130. These components are connected by an internal communication line 146, such as an internal bus.

Moreover, ~~to the switch 100,~~ a management device 160 is connected to the switch 100 via a management console or a port 161 of the Ethernet (registered trademark). In addition, real volumes of storage devices 180 or plural cooperative servers 150 for performing functions, such as backup of a virtual volume provided by the switch 100 in cooperation with the switch 100, are connected to the switch 100 via the connection

interfaces (Ether MACs) 110.

The cooperative server 150 includes: a CPU 151; a main memory (MEM) 152 storing programs and a table 153 such as a volume management table; an Ethernet (registered trademark) (Ether) 156; and a hard disk 157. These components are connected by an internal communication line 155 such as an internal bus via an internal bus control 154.

Fig. 2 is a diagram showing an example of a structure of programs and tables which are stored in the main memory 142 in the switch manager 140 of the switch 100.

A program area 201 of the main memory 142 includes: an operating system 203; a path control protocol 204; real/virtual volume management 207; device control 209; processing of packet for switch 211; and data transfer processing 212. A table area 143 includes: a routing table 205 and a filtering table 206 which are related to the path control protocol 204; a volume management table 208 related to the real/virtual volume management 207; a device management table 210 related to the device management 209; and a unit usage state management table 213. The device management table 210 is a table having recorded therein types, structure information, performance information, and the like of connected devices and incorporated devices. Device management tables in Figs. 4 and 5 are similar [redacted].

Fig. 3 is a diagram showing an example of a structure of programs and tables stored in the main memory 127 and the

memory 123 (125) in the routing control 120 of the switch 100.

The main memory 127 includes: an operating system 301; processing of packet for which hardware processing is impossible 303; and communication processing with the switch manager 302. The memory 125 includes: a routing table 304 related to the communication processing with the switch manager 302; a filtering table 305; and a volume management table 306.

Fig. 4 is a diagram showing an example of a structure of programs and tables stored in the main memory 152 of the cooperative server 150.

A program area 401 of the main memory 152 includes: an operating system 402; real/virtual volume management 403; backup function processing 404; device management 406; and communication processing 408 for communicating with the switch or the management device. The table area 153 includes: a volume management table 405 related to the real/virtual volume management 403 and the backup function processing 404; and a device management table 407 related to the device management 406.

Fig. 5 is a diagram showing an example of a structure of programs and tables stored in the management device 160.

A program area 502 of a main memory 501 includes: an operating system 504, device management 508 and real/virtual volume management 506 as storage management software 505; and communication processing 510 for communicating with the switch

or the cooperative server. A table area 503 includes a volume management table 507 related to the real/virtual volume management 506; and a device management table 509 related to the device management 508.

The volume management tables 208, 306, 405, and 507 shown in Figs. 2 to 5 include a unit management table 600, a logical unit management table 701, and a virtual volume management table 702 shown in Figs. 6 to 8.

An administrator of the computer system shown in Fig. 1 creates or updates the volume management table 507 with the management device 160 and distributes it to the volume management table 208 of the switch manager 140 of the switch 100 and the volume management table 405 of the cooperative server 150. The switch manager 140 distributes it to the volume management table 306 of the routing control 120.

Fig. 6 is a diagram showing an example of a structure of ~~a~~ unit management table 600.

The storage device 180 includes: plural connection ports 601 for making connection with the computer 170 or the switch 100; and disk adapters 603 for accessing plural disks 604. In the case in which there are ~~the~~ plural connection ports 601 and ~~the~~ plural disk adapters 603, each connection port and each disk adapter ~~are~~ is connected by an internal switch 602.

The unit management table 600 includes plural entries 605.

The entry 605 for each disk includes information, such as a unit ID, a media access control (MAC) address, an IP address, a port ID, a disk number, the number of blocks (in order to simplify explanation, one megabyte is assumed to be one block), and capacity.

Fig. 7 is a diagram showing an example of structure of a logical unit management table and a virtual volume management table.

A logical unit management table 701 includes plural entries 701A.

The entry 701A for each logical unit includes information, such as a logical unit (LU) ID, the unit ID shown in Fig. 6, a start logical block address (LBA), an end LBA, a size, and a state.

A virtual volume management table 702 includes plural entries 703.

The entry 703 for each virtual volume includes information, such as a virtual volume ID, a pair of constituent logical units (LUs), and a size and information indicating the propriety of transfer. The propriety of transfer is "prohibited" in an entry 703A and is "possible" in an entry 703B.

A virtual volume 704, shown as an example, is constituted as a combination of logical units with logical unit IDs of VLU00 and VLU01 as indicated by the entry 703A of the virtual volume table 702, and the propriety of transfer is "prohibited".

Fig. 8 is a diagram showing an example of a structure of a logical unit management table and a virtual volume management table as shown in Fig. 7.

A virtual volume 804 is constituted as a combination of logical units with logical unit IDs of VLU10 and VLU11 as indicated by an entry 703B of the virtual volume table 702, and the propriety of transfer is "possible".

The virtual volume of Fig. 7 includes disks 604 in the same storage device 180, and the virtual volume of Fig. 8 includes disks 604 in different storage devices 180. The data transfer method of the present invention performs data transfer in a case as shown in Fig. 8.

Fig. 9 is a diagram which will be referred to for explaining an outline of the processing in the case in which data transfer is performed in the virtual volume 804 of Fig. 8.

Fig. 10 is a diagram showing an example of a processing sequence for obtaining an opportunity for the data transfer processing 212 of the switch manager 140 to perform data transfer.

In the processing sequence of Fig. 10, when the data transfer processing 212 selects a virtual volume, for example, the virtual volume 804 (step 1001), next, the data transfer processing 212 judges whether or not data transfer is prohibited for the selected virtual volume (step 1008). If a result of the judgment indicates that data transfer is prohibited, the

data transfer processing 212 does not perform data transfer. If data transfer is not prohibited, the data transfer processing 212 proceeds to step 1002.

In step 1002, the data transfer processing 212 checks a constituent LU. In the case of the selected virtual volume 804, it is seen from the entry 703B of the virtual volume management table 702 that a constituted logical unit is a combination of VLU10 and VLU11.

In step 1003, the data transfer processing 212 judges whether or not all IP addresses of the constituent LU coincide with each other. If all the IP addresses of the constituent LU coincide with each other, the data transfer processing 212 does not perform data transfer, and, if not, the processing proceeds to step 1004. In the case of the virtual volume 804, unit IDs are RUA0100 and RUB0000, respectively, according to entries 801A1 and 801A2 of the logical unit management table 701. Therefore, since all IP addresses do not coincide with each other according to the unit management table 600, the data transfer processing 212 proceeds to step 1004.

In step 1004, the data transfer processing 212 selects a constituent LU and proceeds to step 1005. Here, in the case of the virtual volume 804, it is assumed that a logical unit of RUA0100 is selected.

In step 1005, the data transfer processing 212 judges whether or not an unused area in a real unit, to which data

can be transferred from another constituent LU, exists in a memory device of the selected constituent LU. If an unused area exists, the data transfer processing 212 proceeds to step 1007, and, if not, the processing proceeds to step 1006. In the case of the virtual volume 804, since a logical unit RUA0100 is selected, and, as a result of the judgment of step 1005, an area to which data can be transferred from another constituent logical unit RUB0000 exists, the data transfer processing 212 proceeds to step 1007.

In step 1007, the data transfer processing 212 performs data transfer processing. In the case of virtual volume 804, the data transfer processing 212 transfers data of VLU11 in RUB0000 to RUA0100.

In step 1006, if there is another constituent LU, the data transfer processing 212 returns to step 1004, and, if not, the processing does not perform data transfer.

In the processing sequence shown in Fig. 10, data transfer may not be performed (in the case of YES in step 1003) or it may not be able to be performed (in the case of NO in step 1006).

In addition, in the processing sequence shown in Fig. 10, the start of the processing sequence may be instructed by an administrator from the management device 160 or the like, or it may be instructed according to a schedule of the storage management software 505.

Fig. 11 is a diagram showing an example of a processing sequence of the data transfer processing 1007.

Fig. 12 is a diagram showing an example of a table 1200 for managing a state of a logical unit in a virtual volume. In the figure, a state 1206 is idle when the virtual volume is not involved in data transfer.

In the processing sequence for obtaining an opportunity for performing data transfer shown in Fig. 10, since there is a transfer destination to which data can be transferred in the unit RUA0100, the data transfer processing 1007 sets a logical unit VLU12 having the same size as the logical unit VLU11, as shown in an entry 1209. In this case, the data transfer processing 1007 sets a state field 1206 to indicate transfer destination (step 1101).

In the logical unit VLU11 of a transfer source, the data transfer processing 1007 sets the state field 1206 to indicate transfer source, as shown in an entry 1208 (step 1102).

The processing up to this point corresponds to (1) LU preparation in Fig. 9.

Next, the data transfer processing 1007 transfers data from the logical unit VLU11 of the transfer source to the logical unit VLU12 of the transfer destination (step 1103) and changes a logical unit constituting a virtual volume from the logical unit VLU11 of the transfer source to the logical unit VLU12 of the transfer destination (step 1105).

The processing up to this point is (2) data transfer and (3) logical unit switching in Fig. 9.

Next, in the logical unit VLU12 of the transfer destination, the data transfer processing 1007 sets the state field 1206 to idle, as shown in an entry 1210, and sets the propriety of transfer of a virtual volume VVOL02k to prohibited (step 1106), deletes the logical unit VLU11 of the transfer source (step 1107), and deletes data in the logical unit VLU11 of the transfer source (step 1108).

The processing up to this point is (4) deletion of the logical unit VLU11 and (5) deletion of data in the logical unit VLU11 of Fig. 9.

Note that, ~~without~~ performing (4) deletion of the logical unit VLU11 and (5) deletion of data in the logical unit VLU11, the data transfer processing 1007 may keep the logical unit VLU11 and the data in the logical unit VLU11 for backup temporarily or for a designated period.

Since the data transfer processing (step 1103) of Fig. 11 is also required to process I/O of Read and Write during data transfer, as shown in Fig. 13, a block bitmap table 1301 for managing incompletion of data transfer for each block unit is provided.

The data transfer processing transfers data ~~by a~~ unit of block from the logical unit VLU11 of the transfer source to the logical unit VLU12 of the transfer destination (step 1104A), updates the block bitmap table 1301 (step 1104B), and repeats the same processing until transfer of all data is

completed (step 1104C).

The I/O processing during data transfer (step 1109) performs processing as shown in a processing matrix 1302 in Fig. 13.

I/O processing 1303 for a data non-transfer area is directly applied to a pertinent memory area, and I/O processing 1304 or 1305 for a data transfer area is applied to a memory area of a logical unit of a data transfer source or a data transfer destination according to the state of the block bitmap table 1301.

In the case in which the structure of the volume management table 208 has been changed by the data transfer processing 212 of the switch manager 140 shown in Figs. 10 to 13, the volume management table 306 of the routing control 120, the volume management table 507 of the management device 160, and the volume management table 405 of the cooperative server 150 are also changed.

Next, a second embodiment of the present invention will be described with reference to Figs. 14 to 16.

In the second embodiment, a case will be described in which data transfer is performed during operation from one or more memory areas already constituting a virtual volume to another memory area, such that the number of memory areas with a relatively small capacity not used in the structure of the virtual volume is reduced.

Fig. 14 is a diagram showing an example of the structure

of a logical unit management table and a virtual volume table, which is the same as the example ~~of a structure~~ shown in Fig. 7.

Fig. 15 is a diagram showing an example of a processing sequence for obtaining an opportunity for performing data transfer.

Fig. 16 is a diagram showing an example of a unit usage state management table 213 for managing the easiness of constitution of a unit.

As an example, the easiness of constitution of a unit is "low" in the case in which the number of vacant blocks is less than 7,500, "medium" in the case in which the number of vacant blocks is less than 12,500, and "high" in the case in which the number of vacant blocks is 12,500 or more.

In the processing sequence of Fig. 15, first, data transfer processing selects a unit (step 1501). As an example, it is assumed that the data transfer processing selects a unit RUA0001 (entry 604A2 of Figs. 6 and 7).

Next, the data transfer processing checks the easiness of constitution of a memory area (step 1502) and proceeds to step 1503.

In step 1503, the data transfer processing judges whether or not an area with low easiness of constitution exists in the selected unit, and, if ~~the~~ such an area does not exist, data transfer is not performed. If ~~the~~ such an area exists, the data transfer

processing proceeds to step 1504. In the case of the example, the data transfer processing checks easiness of constitution of each memory area of the unit RUA0001. Since it is seen that there are memory areas of "medium" and "low" in a field of easiness of constitution 1605 in an entry with a unit ID of RUA001 of Fig. 16, the data transfer processing performs processing, which is the same as that of the processing step 1004 and subsequent steps shown in the processing sequence of Fig. 10, in processing of processing step 1504 and subsequent steps of Fig. 15.

However, the second embodiment, ~~shows the~~ case in which, if data of a logical unit of a transfer source cannot be directly transferred to a logical unit of a final transfer destination, the data is transferred to at least one indirect logical unit, ~~once~~ and, then, transferred to the final logical unit of the transfer destination.

In Fig. 14, (1) the data transfer processing creates an indirect logical unit VLU02 in a unit RUA000X, (2) after copying data from the logical unit VLU01 of the transfer source in a unit RUA0000 to VLU02 in the unit RUA000X, copies the copied data of VLU02 in the unit RUA000X to the unit RUA0000, which is a final transfer destination, and sets a new logical unit VLU01 to this copied data, (3) switches the original logical unit VLU01 to the new logical unit VLU01, (4) deletes the indirect logical unit VLU02, and (5) deletes data in the logical unit VLU02.

In Fig. 16, the unit RUA000X is changed as shown in an entry 1607 in (1) of Fig. 14, the units RUA0001 and RUA000X are changed as shown in an entry 1608 in (4) of Fig. 14, and the unit RUA0001 finally has high easiness of constitution.

Next, a third embodiment of the present invention will be described with reference to Figs. 17 and 18.

In the third embodiment, a case in which main components of a switch are constituted redundantly will be described, and a distribution method of a table from a management device and a synchronization method of a table of a switch control will be described.

Fig. 17 shows an example of a structure of a computer system in the case in which main components of the switch 100 are constituted redundantly.

In Fig. 17, the routing control 120, the crossbar switch 130, and the switch manager 140 are constituted redundantly.

As provided in the first and second embodiments, the unit management table 600, the logical unit management table 701, and the virtual volume management table 702 are distributed from the management device 160 to switch managers 140A and 140B as indicated by a flow of information 1802 in Fig. 18 and are stored in main memories 142A and 142B.

The switch managers 140 constituted redundantly may operate as an active system and a standby system, or both the switch managers 140 may be active systems to share processing.

In the case in which one system of the switch managers 140 constituted redundantly fails, it is switched to the standby system in the case of the active/standby systems; and, in the case in which both the switch managers 140 are the active systems, degenerate operation is performed.

The tables 143 held by the switch managers 140 constituted redundantly are synchronized, as indicated by a flow of information 1801, whereby it becomes possible to switch the tables 143 at a relatively high speed in the case in which one system fails.

Next, a fourth embodiment of the present invention will be described with reference to Fig. 19.

In the fourth embodiment, a case will be described in which data transfer is performed when a switch manager of a switch includes a hard disk.

Fig. 19 is a diagram showing an example of a structure in which the switch manager 140 of the switch 100 includes a hard disk 1901.

As ~~shown~~ provided in the first and second embodiments, the hard disk 1901 is also treated as a part of the unit management table 600 (see Fig. 6) to thereby make it possible to create a logical unit and constitute a virtual volume.

As an example, a unit management table 1903 adopts a structure as shown in an entry 1904.

An indirect logical unit in transferring data is created

in the unit RUS0000, whereby the logical unit can be used in the case in which the memory area of the storage device 180 (see Fig. 1) decreases relatively, and it becomes possible to reduce the number of times data, at the time of data transfer, passes a switch.

According to the present invention, a virtual volume can be constituted by memory areas of the relatively small number of memory devices to reduce relay processing load on a relay device.

In addition, unused memory areas with a relatively small capacity can be reduced, and, as a result, continuous vacant memory areas with a relatively large capacity can be increased.